UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/731,060	12/07/2000	Edward Colles Nevill	550-192	1332

23117 7590 12/05/2008

NIXON & VANDERHYE, PC
901 NORTH GLEBE ROAD, 11TH FLOOR
ARLINGTON, VA 22203

EXAMINER	
ZHEN, LI B	

ART UNIT	PAPER NUMBER
2194	

MAIL DATE	DELIVERY MODE
12/05/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte EDWARD COLLES NEVILL and
ANDREW CHRISTOPHER ROSE

Appeal 2008-3073
Application 09-731,060[1]
Technology Center 2100

Decided: December 5, 2008

Before HOWARD B. BLANKENSHIP, JAY P. LUCAS and THU A. DANG, *Administrative Patent Judges*.

LUCAS, *Administrative Patent Judge*.

DECISION ON APPEAL

---

[1] Application filed December 7, 2000. Appellant claims the benefit under 35 U.S.C. § 119 of various UK applications, filed 10/5/2000 and 11/20/2000. The real party in interest is ARM Limited, a U.K Corporation.

STATEMENT OF CASE

Appellants appeal from a final rejection of claims 1 to 16 under authority of 35 U.S.C. § 134. The Board of Patent Appeals and Interferences (BPAI) has jurisdiction under 35 U.S.C. § 6(b). An Oral Hearing was held on November 5, 2008.

Appellants' invention relates to an apparatus (and method) for processing machine independent coding (e.g. Java bytecode) using hardware based instruction execution when possible, and automatically resorting to software interpretation when necessary. The hardware unit monitors when it is safe to trigger a scheduled multithreading operation. In the words of the Appellants:

> A processing system provides both hardware instruction translation (68) and software instruction interpretation (84) mechanisms for supporting high level program instructions. All of the program instructions are supplied to the hardware translation unit (68) which forwards those instructions it does not itself support to the software interpretation mechanism (84). By routing all program instructions through the hardware translation unit (68), the hardware translation unit (86) is able to monitor when it is appropriate and safe to trigger a scheduling operation for controlling multitasking or multithreaded operations. The scheduling operations may be triggered based upon a count of executed program instructions or by using a timer based scheduling approach with the timer signal being qualified by a signal indicating an appropriate point within the cycle of execution of program instructions.

(Abstract; Spec. 33).

Claim 1 is exemplary:

1. Apparatus for processing data operable to execute operations specified in a stream of program instructions, said apparatus comprising:

(i) a hardware based instruction execution unit operable to execute program instructions; and

(ii) a software based instruction execution unit operable to execute program instructions; wherein

(iii) program instructions to be executed are sent to said hardware based execution unit for execution;

(iv) program instructions received by said hardware based execution unit for which execution is not supported by said hardware based execution unit are forwarded to said software based execution unit for execution with control being returned to said hardware based execution unit for a next program instruction to be executed; and

(v) said hardware based execution unit includes scheduling support logic operable to generate a scheduling signal for triggering a scheduling operation to be performed between program instructions for managing scheduling between threads or tasks irrespective of whether a preceding program instruction was executed by said hardware based execution unit or said software based execution unit.

The prior art relied upon by the Examiner in rejecting the claims on appeal is:

Evoy	US 5,937,193	August 10, 1999
Gee	US 6,374,286	April 16, 2002

REJECTIONS

Claims 1 to 16 stand rejected under 35 U.S.C. § 103(a) for being obvious over Evoy in view of Gee.

The claims are argued together. Appellants contend that the claimed subject matter is not rendered obvious by Evoy in combination with Gee for failure of the combination of references to teach claimed limitations. The Examiner contends that each of the claims is properly rejected.

Rather than repeat the arguments of Appellants or the Examiner, we make reference to the Briefs and the Answer for their respective details. Only those arguments actually made by Appellants have been considered in this opinion. Arguments which Appellants could have made but chose not to make in the Briefs have not been considered and are deemed to be waived.

We affirm the rejections.

ISSUE

The issue is whether Appellants have shown that the Examiner erred in rejecting the claims under 35 U.S.C. § 103(a). The issue turns on whether the references Evoy and Gee teach or suggest the return of control to the hardware execution unit after it is forwarded to the software execution unit, whether scheduling between processing threads is performed in the claimed manner, and other related claimed limitations.

FINDINGS OF FACT

The record supports the following findings of fact (FF) by a preponderance of the evidence.

1. Appellants have invented a system for executing device independent instruction code using both fast hardware processing and, when the hardware cannot process the instructions, more flexible software execution. (App. Br. 3, top). After complex bytecode (instructions) have

been processed using the software interpreter, control is returned to the hardware based execution. (App. Br. 4, bottom). The hardware execution unit contains logic, in the form of a program counter, to help schedule branching at safe points in the processing when interruptions can be handled. (Spec. 26, top). This logic also helps manage scheduling between threads in multithreaded processing. (Spec. 27, top).

2. The reference Evoy teaches a circuit (and method) for executing platform independent program code. (Col. 3, l. 43). The processor is configured to operate in two modes; the native mode of the processor and in platform independent (Java) mode, where the Java bytecodes are translated into the native instructions. (Col. 4, ll. 53 to 63). Normally both modes are executed by fast hardware processing, but when no native instruction exists to process the bytecode, a software interpreter is called to interpret the bytecode. (Col 5, ll. 58-67).

3. The reference Gee, in a device that consists of multiple Java processors that process Java bytecode, teaches the use of a priority based scheduling policy to avoid interrupting the execution of a thread in the middle of processing that thread. (Col. 21, ll. 25-55).

PRINCIPLES OF LAW

Appellants have the burden on appeal to the Board to demonstrate error in the Examiner's position. See *In re Kahn*, 441 F.3d 977, 985-86 (Fed. Cir. 2006) ("On appeal to the Board, an applicant can overcome a rejection [under § 103] by showing insufficient evidence of *prima facie* obviousness or by rebutting the *prima facie* case with evidence of secondary

indicia of nonobviousness.") (quoting *In re Rouffet*, 149 F.3d 1350, 1355 (Fed. Cir. 1998)).

"What matters is the objective reach of the claim. If the claim extends to what is obvious, it is invalid under § 103." *KSR Int'l Co. v. Teleflex, Inc.*, 127 S. Ct. 1727, 1742 (2007). To be nonobvious, an improvement must be "more than the predictable use of prior art elements according to their established functions." *Id.* at 1740.

ANALYSIS

From our review of the administrative record, we find that Examiner has presented a prima facie case for the rejections of Appellants' claims under 35 U.S.C. § 103. The prima facie case is presented on pages 3 to 10 of the Examiner's Answer. In opposition, Appellants present a number of arguments.

Arguments with respect to the rejection of claims 1 to 16 under 35 U.S.C. § 103

Appellants contend that Examiner erred in rejecting claims 1 to 16 under 35 U.S.C. § 103(a). Appellants first argument is that "contrary to the Examiner's assertion, [col.] 7: [lines] 8-16 of Evoy does not disclose the claimed feature where control is returned to the hardware based execution unit for a next program instruction to be executed." (App. Br. 9, bottom). Appellants contend that "Evoy never teaches returning control [to] the hardware-based execution unit, corresponding to the translation circuit 50 used in the platform-independent mode of Evoy, after a bytecode has been

forwarded to the software based execution unit (the software interpreter) for execution." (App. Br. 10, middle).

Evoy teaches that the hardware processor is designed to handle native code or platform-independent code (Java bytecode). The Java bytecode is hardware translated to native code by translation circuit 50 and associated components to be processed as native code. (*See* FF #2). However, occasionally a section of the Java bytecode is beyond the ability of the translation circuitry to translate, and the coding is then sent to a software interpreter to be processed. (Col. 7, ll. 14-17).

Appellants would have us understand that, after the exception processing by the software unit, Evoy does not return to the normal hardware execution for the processing of the rest of the computer program. Appellants appear to read the paragraph on incrementing the address counter to process the next section of code as only applying to native instructions hardware translated by the hardware circuitry while in the platform-independent code. We do not find this as a reasonable reading of Evoy. By Appellants' reading of the reference, processing would stop after the call to the software interpreter. (Appellants do not explain what happens after the call to the software interpreter by their reading of the reference.) We find it more reasonable to read Evoy as teaching that after the call to the software interpreter to process the difficult section of Java bytecode processing resumes in the normal execution of the instructions in the program. The next line of program instructions in Evoy is directly executed, as claimed, and if the next instruction is platform-independent, then it is handled by the

hardware units or software units depending on the instruction complexity, as described in the patent (Col. 4, l. 53 *ff*).

At Oral Hearings, the Appellants made a point of indicating that the claims require that a complicated line of bytecode that was being interpreted by the software interpreter not be interrupted. Thus, if a program instruction is software interpreted into three native instructions, for example, the claims require a return to execution by the hardware unit after each native instruction is generated. (Oral Hearings, Nov. 5, 2008). However, we note that the claims only require a return to the hardware unit after each *program instruction* is executed, not the resultant interpreted native instructions. (*See* claim 1 above). As mentioned above, this feature is taught by Evoy.

We thus do not find Appellants' first contention convincing of Examiner error.

Appellants next contend that Gee does not teach the scheduling policy "like that claimed". (App. Br. 13, middle). More specifically, Appellants state "[t]he issue of managing scheduling 'irrespective of whether a preceding program instruction was executed by the hardware based execution unit or the software based execution unit' does not arise in Gee because Gee does not disclose both the hardware-based and software-based execution units." (App. Br. 13, bottom to 14, top). Gee teaches a multi-processor computer that executes multiple threads while processing Java coding. As pointed out by the Examiner, the processing in Gee has both hardware and software components executing the bytecode. (*See* Ans. 5, bottom to 6, bottom). Gee specifically mentions scheduling support logic managing the threads to trigger operations between threads, not in the

middle of processing a single thread – Rule 1. (Col. 21, ll. 28-58). Notice at line 38 that executing threads have the highest priority, thus staying any interruption until the thread is fully executed.

In view of the teachings of Gee, combined with the basic teaching of Evoy, we decline to find error in the Examiner's rejection on this point.

Appellants' arguments concerning the dependant claims were adequately answered by the Examiner, whose position we adopt. (Ans. 7).

Appellants' allege a failure of motivation to combine noting that the references were addressed to a different purpose. (App. Br. 15, middle). However, this issue was addressed by the Supreme Court in a recent case: "It is common sense that familiar items may have obvious uses beyond their primary purposes, and a person of ordinary skill often will be able to fit the teachings of multiple patents together like pieces of a puzzle." *KSR Int'l Co. v. Teleflex, Inc*., 127 S. Ct. 1727, 1732 (2007). We thus decline to find error in the combination.

CONCLUSION OF LAW

Based on the findings of facts and analysis above, we conclude that the Examiner did not err in rejecting claims 1 to 16 under 35 U.S.C. § 103.

DECISION

The Examiner's rejection of claims 1 to 16 under 35 U.S.C. § 103 is Affirmed.

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv).

AFFIRMED

msc

NIXON & VANDERHYE, PC
901 NORTH GLEBE ROAD, 11TH FLOOR
ARLINGTON VA 22203